A Multiple Error LMS Algorithm and Its Application to the Active Control of Sound and Vibration

STEPHEN J. ELLIOTT, MEMBER, IEEE, IAN M. STOTHERS, AND PHILIP A. NELSON

Abstract—An algorithm is presented to adapt the coefficients of an array of FIR filters, whose outputs are linearly coupled to another array of error detection points, so that the sum of all the mean square error signals is minimized. The algorithm uses the instantaneous gradient of the total error, and for a single filter and error reduces to the “filtered x LMS” algorithm. The application of this algorithm to active sound and vibration control is discussed, by which suitably driven secondary sources are used to reduce the levels of acoustic or vibrational fields by minimizing the sum of the squares of a number of error sensor signals. A practical implementation of the algorithm is presented for the active control of sound at a single frequency. The algorithm converges on a timescale comparable to the response time of the system to be controlled, and is found to be very robust. If the pure tone reference signal is synchronously sampled, it is found that the behavior of the adaptive system can be completely described by a matrix of linear, time invariant, transfer functions. This is used to explain the behavior observed in simulations of a simplified single input, single output adaptive system, which retains many of the properties of the multichannel algorithm.

I. INTRODUCTION

The active control of sound or vibration involves the introduction of a number of controlled “secondary” sources driven so that the field generated by these sources interferes destructively with the field caused by the original “primary” source [1]–[3]. The extent to which such destructive interference is possible depends on the geometric arrangement of the primary and secondary sources and their environment, and on the spectrum of the field produced by the primary source [4]. In broad terms, considerable cancellation of the primary field can be achieved if the primary and secondary sources are positioned within half of a wavelength of each other at the frequency of interest [5]. Active methods of control are thus best at attenuating low-frequency sound, which complements more conventional passive methods of control since these tend to work best at higher frequencies [1].

One form of primary sound or vibration field which is of particular importance in practice is that produced by rotating or reciprocating machines. The waveform of the primary field in these cases is nearly periodic, and since it is generally possible to directly observe the action of the machine producing the original disturbance, the fundamental frequency of the excitation is generally known. Each secondary source can then be driven at each harmonic via a controller which adjusts the amplitude and phase of a reference signal whose frequency is arranged to be at multiples of this known fundamental frequency. It is often desirable to make this controller adaptive. This is usually because the frequency or spatial distribution of the primary field changes with time, and the controller is required to track these changes. A more difficult adaptive task has to be performed when the response of the system to be controlled to a given secondary excitation also varies with time. In this case, an algorithm which simultaneously performs identification and control must be implemented. This problem is not addressed in this paper in which it is assumed that the response of the system to be controlled does not change during adaptation, and can be measured during an identification phase prior to control.

In order to construct a practical adaptive controller, some measurable error criterion must be defined which the controller is required to minimize. Although the minimization of total radiated power, under acoustic free field conditions, or total acoustic potential energy, for enclosed sound fields, have been proposed in theoretical formulations, to determine the best possible attenuation which can be achieved with an active control system [4], these quantities are generally not practically measurable. One error criterion which can be directly measured is the sum of the squares of these quantities.

The signal processing problem is then to design an adaptive algorithm to minimize the sum of the squares of a number of sensor outputs by adjusting the magnitude and phases of the sinusoidal inputs to a number of secondary sources. Section II of this paper puts forward an algorithm to perform this task, assuming that a sampled system is used and that digital filters are used to implement the controller. The section continues with a discussion of the optimum least mean sum of squares solution to this problem.
Section III discusses the use of the algorithm in a practical active sound control experiment. If the fundamental excitation frequency is known, considerable computation savings can be made by arranging for the sample rate to be an integer multiple of this frequency, thus ensuring synchronous sampling [6], [7]. Under these conditions, it is found that the adaptive algorithm behaves like a linear time invariant system. The matrix of transfer functions relating the controlled outputs of the sensors to their original uncontrolled outputs is described under these conditions in the Appendix. This equivalent transfer function approach is then used to analyze the behavior of a single input, single output system, in Section IV, since such an analysis throws considerable light on the behavior of the multiple error algorithm. Some possible extensions and modifications to the algorithm are then discussed in Section V.

II. A MULTIPLE ERROR LMS ALGORITHM

A. Derivation

Let the sampled output of the $l$th error sensor be $e_l(n)$, which is equal to the sum of the “desired” signal from this sensor, $d_l(n)$, due to the primary source operating alone, and an output due to each of the actuators. Let the sampled input to the $m$th actuator be obtained by filtering the reference signal $x(n)$ using an adaptive FIR controller whose $i$th coefficient at the $n$th sample is $w_{mi}(n)$. Let the transfer function between this input and the output of the $l$th sensor be modeled as a $J$th-order FIR filter, whose $j$th coefficient is $c_{lmj}$, so that

$$e_l(n) = d_l(n) + \sum_{m=1}^{M} \sum_{j=0}^{J-1} c_{lmj} \sum_{i=0}^{I-1} w_{mi}(n-j)x(n-i-j).$$

(1)

It is assumed that there are $L$ sensors and $M$ actuators, and that $L \geq M$. Let the total error $J$ be defined as

$$J = E\left\{ \sum_{l=1}^{L} e_l^2(n) \right\}$$

(2)

where $E\{ \cdot \}$ denotes an expectation value. If the reference signal $x(n)$ is at least partly correlated with each $d_l(n)$, it is possible to reduce the value of $J$ due to the primary source alone, by driving the secondary sources with a filtered version of the reference signal, as indicated above.

It is physically clear that the total error will be a quadratic function of each of these filter coefficients (although this is also demonstrated analytically below). The optimum set of filter coefficients required to minimize $J$ may thus be evaluated adaptively using gradient descent methods. The differential of the total error with respect to one coefficient is

$$\frac{\partial J}{\partial w_{mi}} = 2E\left\{ \sum_{l=1}^{L} e_l(n) \frac{\partial e_l(n)}{\partial w_{mi}} \right\}. \quad (3)$$

Assuming for the moment that each $w_{mi}$ is time invariant, differentiating (1) with respect to one of these coefficients gives

$$\frac{\partial e_l(n)}{\partial w_{mi}} = \sum_{j=0}^{J-1} c_{lmj} x(n-i-j). \quad (4)$$

This sequence is the same as the one which would be obtained at the $l$th sensor if the reference signal, delayed by $i$ samples, were applied to the $n$th actuator. Let this be equal to $r_{ln}(n-i)$, a filtered reference.

If each coefficient is now adjusted at every sample time by an amount proportional to the negative instantaneous value of the gradient, a modified form of the well-known LMS algorithm is produced [8].

$$w_{mi}(n + 1) = w_{mi}(n) - \alpha \sum_{l=1}^{L} e_l(n) r_{ln}(n-i) \quad (5)$$

where $\alpha$ is the convergence coefficient. This is a form of “stochastic gradient” adaptive algorithm.

For a single input, single output system ($L = M = 1$), this corresponds exactly to the “filtered $x$ LMS” algorithm discussed by Widrow and Stearns [9]. The single error LMS algorithm with a delay in the reference path was originally presented by Widrow in 1971 [20, Fig. 9]. The use of a more general filtering of the reference signal in single input, single output systems has been discussed by Morgan [16], Burgess [10], and Widrow et al. [21]. The assumption of time invariance in the filter coefficients is equivalent, in practice, to assuming that the filter coefficients $w_{mi}$ change only slowly compared to the timescale of the response of the system to be controlled. This timescale is defined by the values of the coefficients $c_{lmj}$.

B. Time Domain Analysis

In order to analytically demonstrate the shape of the error surface, and so determine the optimum, Wiener, set of filter coefficients, it is convenient to consider the case in which the filter coefficients are exactly time invariant. In this case, (1) may be written

$$e_l(n) = d_l(n) + \sum_{m=1}^{M} \sum_{i=0}^{I-1} w_{mi} \sum_{j=0}^{J-1} c_{lmj} x(n-i-j) \quad (6)$$

$$+ \sum_{m=1}^{M} \sum_{i=0}^{I-1} w_{mi} r_{ln}(n-i) \quad (7)$$

where the filtered reference signal $r_{ln}(n)$ is defined as above. This equation may be written as

$$e_l(n) = d_l(n) + r_l^T w \quad (8)$$
where
\[ r^T_l = [r_1(n), r_1(n-1), \ldots, r_1(n-L), \ldots] \]
\[ \begin{align*}
w^T &= [w_{10}, w_{11}, \ldots, w_{M0}, \ldots, w_{M1}] \]
\[ \begin{bmatrix}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array{false}
III. AN ACTIVE SOUND CONTROL EXPERIMENT

The multiple error LMS algorithm has been programmed in assembly language on a Texas Instruments TMS 32010 microprocessor. This device was used in a programmable real-time signal processor developed at the ISVR, which has six analog outputs and eight analog inputs [13]. This processor was used to control the outputs from four microphones by providing the inputs to two (secondary) loudspeakers, as indicated in Fig. 1. The two secondary loudspeakers (KEF B200G) were placed about 300 mm either side of another loudspeaker (KEF B139B) which supplied the primary excitation. The loudspeakers were placed about 1.5 m from the floor against one wall of a laboratory about 12 m by 8 m wide by 3.5 m high. The laboratory had an acoustic "reverberation time" at this frequency (i.e., the time for the pressure amplitude to fall by a factor 1/e) is about 170 ms. The four microphones were placed about 2 m from the loudspeakers and were distributed around them at various heights between 1 and 2 m from the floor.

The sampling rate supplied to the processor was divided down by a factor of four to provide an internal reference signal \( x(n) \), and a sinusoidal analog output to drive the loudspeaker acting as the primary source. The frequency of the reference signal was about 100 Hz. The transfer functions from each secondary loudspeaker to each error microphone were modeled with two point filters at this frequency during an initialization phase of the program. These filters were subsequently used to generate the filtered reference signals \( r_m(n) \). The convergence coefficient in the algorithm was adjusted until it was judged that the convergence rate was fastest. The program in the TMS32010 allowed the values of the filter coefficients and the error signals to be stored away in external memory. This memory was subsequently interrogated and used to generate Fig. 2, which shows the total error, computed using the expression for \( J(n) \) below, together with the filter coefficient trajectories over 350 samples from the instant the adaption is started.

\[
J(n) = \left[ \sum_{i=1}^{4} e_i^2(n) + \sum_{i=1}^{4} e_i^2(n - 1) \right]/2. \tag{18}
\]

Since the sample rate is 400 Hz, 350 samples correspond to a timescale of 875 ms. The primary field is arranged to be constant at the time of adaption. It can be seen that although the filter coefficients start to change at once, the total error only changes after about 10 samples, due to the delays in the secondary paths. These delays are of about 25 ms or 10 sample periods duration, and are not only due to the acoustic propagation time (about 6 ms) but also delays through the analog antialiasing and reconstruction filters, which are in fact dominant in this case.

In this experiment, it can be seen from Fig. 2 that the system has substantially converged in about 100 samples. This corresponds to an actual time of 250 ms which should be compared to the 25 ms overall delay and the 170 ms time constant of the system being controlled.

The results presented in Fig. 2 are very similar in form to computer simulations of a two source, four sensor control system reported in [11]. In this simulation, the average delay in the model of the system to be controlled was 14 samples, and the "reverberant" property of the system was modeled by a simple recursion. The experimental results for different values of convergence coefficient (\( \alpha \)) also exhibit the behavior found in the simulations: if \( \alpha \) is smaller than that used to generate Fig. 2, the total error still decays monotonically, but more slowly. If \( \alpha \) is larger than that used to generate Fig. 2, the total error begins to ring as it converges. The frequency of the ringing is related to the magnitude of the delay in the system to be controlled. If \( \alpha \) is increased even further, these oscillations grow rather than die away and the system becomes unstable.

The algorithm has been found to be very robust to errors in the generation of each of the filtered reference signals \( r_m(n) \). In particular, the algorithm can be made stable even with nearly 90° phase error in these signals, although the convergence parameter must be reduced somewhat to maintain stability in this case. This phase condition is intuitively reasonable in the case of slow convergence, since it implies that the average value of the individual terms in each update equation \( e_i(n) r_m(n - i) \) must be at least of the correct sign for the error to be reduced during adaption and thus retain stability. The robustness of the algorithm is also demonstrated by other simulations which show that the convergence is largely unaffected by the introduction of either considerable uncorrelated observation noise or moderate nonlinearity in the transfer functions relating the sensor outputs to the actuator inputs.

It should be noted that synchronous sampling is not necessary for the algorithm to behave as described below. Indeed, with minor modifications the algorithm has been shown to work using a fixed sample rate and with reference signals whose frequency is swept quite rapidly [12].

IV. THE SINGLE INPUT, SINGLE OUTPUT SYSTEM

A. Simulations of Convergence Behavior

Considerable insight into some aspects of the behavior of the multichannel algorithm may be gained by considering the simplified case of a single input, single output system. Such systems are also worthy of study in their own right since the algorithm reduces to the "filtered x" LMS algorithm [9] in this case.

We first examine the convergence behavior of the algorithm using a simulation in which a single frequency, synchronously sampled, reference signal is used and the secondary path \( C(z) \) is a pure delay. The estimate of the secondary path used to generate the reference signal \( \hat{C}(z) \) is assumed to be perfectly accurate, i.e., \( \hat{C}(e^{j\omega_0}) = C(e^{j\omega_0}) \). Simulations have been performed with more complicated filters modeling \( C(z) \), and these are found to display essentially similar behavior to the simulations presented here. Practical implementations of the algo-
The multiple error LMS algorithm have also been used for the control of plane waves of sound propagating in a duct using a practical arrangement similar to that reported above. The secondary path $C(z)$, in this case, is composed of a delay due to the acoustic propagation time and the analog filters, together with a considerable “reverberant” response due to multiple reflections of sound waves in the duct. The behavior of the practical system is, however, found to be similar to simulations in which only the overall delay is accounted for.

These considerations give some confidence in asserting that it is the overall delay in the error path which principally determines the dynamic behavior of the single input, single output filtered $x$ algorithm. A simulation of the system shown in Fig. 3 has been performed. The delay, of $\Delta$ samples, in the secondary path, has been varied to be integer multiples of the period of the reference signal (i.e., $\Delta = 4D$ where $D$ is an integer). If the delay is an integer number of periods of the reference plus one, two, or three samples, very similar behavior is observed when the delay is only the integer number of reference periods. The squared error signal, averaged over two samples, is shown against the sample number in Fig. 4 with a secondary delay of four samples (1 period of the reference signal) and for three values of the convergence coefficient $\alpha$. It is clear that if $\alpha$ is 0.05, the error converges monotonically, and if $\alpha$ is 0.5, the mean square error develops the characteristic oscillatory behavior observed in the experiments using the multichannel algorithm described above. The convergence time increases as $\alpha$ is decreased for $\alpha < 0.15$, and increases with increasing $\alpha$ for $\alpha \geq 0.2$. The system becomes unstable for $\alpha \geq 0.6$. There is thus an optimum value of convergence coefficient which gives the fastest convergence time. This behavior is further illustrated in Fig. 5. The “convergence time” in this figure is defined as being the time at which the squared error falls below 1 percent of its initial value and does not subsequently rise above this value; it is plotted in units of
Fig. 3. Block diagram of the computer simulation of the single input, single output system.

Fig. 4. Mean square error for single input, single output system with three values of the convergence coefficient: (a) $\alpha = 0.05$, (b) $\alpha = 0.15$, (c) $\alpha = 0.50$.

periods of the reference signal. This convergence time is plotted against the convergence coefficient for a variety of pure delays in the secondary path. The variation of the convergence time with $\alpha$ is not as smooth for $\alpha$ greater than its optimum value as it is for $\alpha$ smaller than its optimum value. This is due to the oscillatory nature of the averaged square error in the former case, which causes the convergence time to be determined by one of the ripples observed in Fig. 4(c) over a range of values of $\alpha$. Thus, the convergence time is nearly constant until $\alpha$ has decreased sufficiently for this ripple to be below 1 percent of the initial error, after which the convergence time is determined by the previous ripple in the response.

It is clear that for very small values of $\alpha$, the behavior becomes similar in all cases, with the convergence time inversely proportional to $\alpha$. This is the same behavior as is observed if no error path is present, $C(z) = 1$, as indicated by the dashed line in Fig. 5, and demonstrates, as expected, that for very slow convergence the dynamics of the error path have no effect. By plotting the logarithm of the convergence coefficient corresponding to the smallest convergence time $\alpha_{opt}$ against the logarithm of the delay in the secondary path, expressed as $D$ periods of the reference signal or $\Delta$ samples, it is found that the relationship between them is well approximated by

$$
\alpha_{opt} \approx \frac{1}{4D} = \frac{1}{\Delta}.
$$

The algorithm is found to be unstable for values of convergence coefficient greater than about three times this value, for all the delays used in Fig. 5.

By plotting a graph of the smallest convergence time $\tau_{\min}$ measured in periods of the reference signal, against the delay $D$, the relationship

$$
\tau_{\min} = 1 + 2.7D
$$

is also found to be an excellent fit to the data. It should be noted that if $D = 0$, the algorithm converges completely in one cycle of the reference which accounts for the first term in (20). Apart from this effect, the convergence time is found to increase very nearly linearly with the delay in the secondary path, with a proportionality constant for the convergence criterion used here of 2.7. This factor is dimensionless since the delay in the error path ($D$) and the optimum convergence time ($\tau_{\min}$) are measured in the same units.

B. Equivalent Transfer Function

The behavior of the single input, single output system, with a synchronously sampled reference signal of frequency $\omega_o$, may be completely represented by an equivalent transfer function, which can be deduced as a special case of that presented in the Appendix. If $L = M = 1$, the output of the single adaptive filter $Y(z)$ can be related to the single error signal $E(z)$ by
\[
\frac{Y(z)}{E(z)} = \frac{I\alpha A}{2} \left[ z \cos(\omega_o - \Phi) - \cos \Phi \right] = G(z)
\]  
(21)

in which the estimate of the secondary path used to derive the filtered reference signal \( \hat{C}(e^{j\omega_o}) \) is given by \( \lambda e^{j\Phi} \).

This represents the transfer function of a linear, time invariant system. So, if \( D(z) \) is the desired signal, \( E(z) = D(z) + C(z) Y(z) \), and

\[
\frac{E(z)}{D(z)} = \frac{1}{1 - C(z) G(z)} = H(z)
\]

(22)

\( H(z) \) is thus the equivalent transfer function between the error output and desired input, and the entire active controller acts as a linear time invariant system between these two signals.

Substituting for \( G(z) \) above and letting

\[
\beta = \frac{I\alpha A}{2},
\]

we obtain

\[
H(z) = \frac{1 - 2z \cos(\omega_o) + z^2}{1 - 2z \cos(\omega_o) + z^2 + \beta C(z) \left[ z \cos(\omega_o - \Phi) - \cos \Phi \right]}.
\]

(23)

This result is important because for any given error path \( C(z) \), it allows the full behavior of the system to be determined analytically. Since no approximations have been made in the derivation of \( H(z) \), the complete, dynamic behavior of the adaptive system is described by this equation.

If we set \( C(z) = 1 \) and \( \Phi = 0 \) in this expression, the equivalent transfer function of the ordinary adaptive noise canceller derived by Glover [14] is obtained. Equation (23) can be used to derive an equivalent transfer function for the simulations performed above. If, for example, the delay in the secondary path is one cycle of the single frequency reference signal used above, so \( C(z) = z^{-4}, \omega_o = \pi/2, \hat{C}(e^{j\omega_o}) = 1, \) and \( I = 2, \) the transfer function becomes

\[
H(z) = \frac{1 + z^{-2}}{1 + z^{-2} - \alpha z^{-6}}.
\]

(24)

The positions of the 6 poles of this transfer function for various values of \( \alpha \) are shown in Fig. 6. It is possible to analytically determine the pole positions in this case by using \( z^{-2} \) as the variable in the denominator and solving the resulting cubic equation. Such an analysis shows that the roots of \( z^{-2} \) are real for \( \alpha < 4/27 \) (0.148) but imaginary for values of \( \alpha \) greater than this value. For small values of \( \alpha \), two of the poles lie on the positive and negative imaginary axis just inside the unit circle, and the other four lie near the origin on the positive and negative real and imaginary axes. As \( \alpha \) is increased, the poles near the unit circle move in and the poles around the origin move out until, at \( \alpha = 0.148 \), the four poles on the imaginary axis break away along the paths indicated in Fig. 6.

As \( \alpha \) is further increased, beyond about 0.6, these four poles migrate outside the unit circle. The values of \( \alpha \) used to calculate the pole positions shown in Fig. 6 are exactly the same as those used to generate the simulation results shown in Fig. 4. The "overdamped" and "underdamped" response of the mean square errors seen in Fig. 4 are clearly explained by the corresponding pole positions in Fig. 6.

A similar diagram of the pole trajectories can be obtained for delays greater than 4 samples, although in this case more than four poles move radially away from the origin as \( \alpha \) is increased. A similar breaking away of the poles on the imaginary axis is still observed, however, and the value of \( \alpha \) which corresponds to this breakaway can be obtained by using standard root locus theory [15]. For \( C(z) = z^{-4D}, \omega_o = \pi/2, \Phi = 0, A = 1, I = 2, \) the equivalent transfer function is

\[
H(z) = \frac{1 + z^{-2D}}{1 + z^{-2D} - \alpha z^{-4D}},
\]

(25)

the poles of which are given by the values of \( z \) satisfying

\[
z^{4D} + z^{4D+2} - \alpha = 0
\]

or

\[
\alpha = z^{4D} + z^{4D+2}.
\]

(26)

The pole break points can be obtained by setting the differential below to zero [15].

\[
\frac{\partial \alpha}{\partial z} = 4Dz^{4D-1} + (4D + 2)z^{4D+1} = 0.
\]

(27)

Therefore, \( z = 0 \) or \( z^2 = -(2D/2D + 1) \). The value of \( \alpha \) corresponding to the latter condition is given from (26) by

\[
\alpha_o = z^{4D}(1 + z^2) = \left[ \frac{2D}{2D + 1} \right]^{2D} \left( \frac{1}{2D + 1} \right).
\]

(28)
For large $D$, this expression limits to:

$$\alpha_o \approx \frac{1}{e} \times \frac{1}{2D} \approx \frac{1}{5.4D}$$

(29)

In fact, this approximation (29) is within about 10 percent of the full expression (28) for $D \geq 2$. This predicted value of $\alpha_o$ is slightly smaller than the value of $\alpha$ corresponding to the fastest convergence rate observed in the simulations above (given by $\alpha_{opt} \approx 1/4D$). This is because $\alpha_o$ corresponds to an error history which has no ripples, whereas a faster convergence time can be obtained if $\alpha$ is slightly increased so that some ripples occur. It is, however, significant how this analysis of the equivalent transfer function can lead to predictions of the behavior of the algorithm which are very close to empirical curves fitted to the results of numerous simulations.

### C. Accuracy of Estimated Secondary Path Transfer Function

Equation (23) also allows the necessary accuracy of the estimated secondary path $\hat{C}$ to be deduced in the limit of slow adaption. If the adaption is assumed to be very slow, i.e., $\beta \to 0$, $W$ becomes nearly time invariant and the physical transfer functions $W$ and $C$ may be reordered as in Fig. 7(a). This is equivalent to the system given in Fig. 7(b), in which $q(n)$ is the new reference signal, which is also a sinusoid at $\omega_o$, and $\epsilon(z) = C(z)/C(z)$, which is the error in the estimate of the error path.

The transfer function in the secondary path has completely disappeared from this diagram so we can set $C(z) = 1$ in (23). However, the error in the estimate of the secondary path remains as $\epsilon(z)$ which is assumed to have a phase response of $\Phi$ at $\omega_o$. The transfer function of this system thus becomes

$$H(z) = \frac{1 - 2z \cos(\omega_o) + z^2}{1 - 2z \cos(\omega_o) + z^2 + \beta(z \cos(\omega_o - \Phi) - \beta \cos \Phi)}$$

$$= \frac{1 - 2z \cos(\omega_o) + z^2}{1 - 2z \cos(\omega_o) + z^2 + \beta(z \cos(\omega_o - \Phi) - \beta \cos \Phi)}$$

(30)

This is a second-order recursive system whose stability can be investigated by examining whether the pole positions are within the unit circle. For small $\beta$, $H(z)$ will have conjugate poles at a distance of $(1 - \beta \cos \Phi)^{1/2}$ from the origin. Since all the terms in $\beta$ are assumed positive, the distance of the pole from the unit circle can only be greater than 1 if $\cos \Phi$ is negative, so the stability condition must be:

$$\cos \Phi > 0.$$  

Therefore,

$$90^\circ > \Phi > -90^\circ.$$  

(31)

This phase condition has been previously suggested by Morgan [16].

An additional condition for stability is that

$$\left|2 \cos(\omega_o) - \beta \cos(\omega_o - \Phi)\right| < 2.$$  

(32)

Since $1 > \cos(\omega_o) > 0$ and $\beta$ is assumed small, this condition must also be satisfied.

The time constant of convergence of an adaptive cancel- ler with a sinusoidal reference but no extra transfer function in the error path is inversely proportional to the $\sin^2$ of the coefficient $C$ at $\omega_o$. Assuming the adaption of the filtered $x$ algorithm is already slow, to account for the dynamic properties of $C$, its convergence is further slowed if $\hat{C}$ is not a good match to $C$ at $\omega_o$. The analysis above indicates that the time con- stant of convergence is slowed down by a factor of $1/\cos \Phi$, where $\Phi$ is the phase difference between $\hat{C}$ and $C$ at $\omega_o$.

Another point worthy of mention is the effect of using multiple frequency components in the reference signal. If the response of the error path at one such frequency $\omega_i$ is $C(e^{j\omega_i})$, and assuming $\hat{C}(z) = C(z)$, then both the error signal and the reference signal at this frequency will be proportional to $|C(e^{j\omega_i})|$. Thus, the update term, $ae(n)r(n-i)$ will be proportional to the square of this modu- lus at frequencies about $\omega_i$. In general, however, the modulus of the response of the filter $C$ at each of the frequencies present in the reference signal will be very different. Since only a single value of the convergence coefficient is used, which applies to all the frequency components in the reference, this must be chosen so the system is stable at the frequency at which the response of $C$ is largest. This will considerably slow down the con- vergence of the algorithm at frequencies where the response of $C$ is small. Such behavior is analogous to that due to the eigenvalue spread of the autocorrelation matrix in the conventional LMS algorithm.

### V. Modification of the Algorithm

#### A. Use of a More General Cost Function

An error function or "cost" function which is widely used in the field of optimal control [15] involves both mean square error terms and terms proportional to the mean square effort. For example, if $y_m(n)$ is the output of the $m$th filter, one cost function which could be used is

$$J_T = E \left\{ \sum_{l=1}^{L} p_l e_l^2 + \sum_{m=1}^{M} q_m y_m^2 \right\}$$  

(33)

where $p_l$ and $q_m$ are the weightings on the individual errors ($e_l^2$) and "efforts" ($y_m^2$), respectively. The differential of this cost function with respect to the $i$th coefficient of the $m$th filter is
ELLIOTT et al.: MULTIPLE ERROR LMS ALGORITHM

Fig. 7. Block diagram for the adaptive system in the case of very slow adaption in (a) reordered physical form and (b) reduced form.

The computer simulation of the multichannel active control system described in [11] was modified to incorporate a simplified form of this cost function. The values of all the error weighting coefficient \( p_i \) were set equal (at unity) and the weighting function for both of the outputs \( q_1 \) and \( q_2 \) were also set equal, at some variable value. If \( q_1 = q_2 = 0 \), the algorithm reduces to that above. As \( q_1 \) and \( q_2 \) were increased, the transient time of the algorithm did not appear to change, but the steady solution, after 600 cycles of the simulation, began to alter. For example, with \( q_1 = q_2 = 10 \) in the simulation, the final mean square error was 1.7 (compared to 0.93 when \( q_1 = q_2 = 0 \), and 3.2 before adaption\) and the final mean square value of the filter outputs was 0.127 (compared to 0.667 when \( q_1 = q_2 = 0 \)). Thus, the algorithm does allow much smaller secondary strengths to be used while still achieving some reductions in the error output.

Cost functions such as these have already been discussed, for example, in the active control of helicopter vibration [17]. Their use would appear to be beneficial whenever there is a possibility of very large source strengths being necessary to achieve very small reductions at the error sensors, leading either to nonlinear behaviour or to increases in the total field away from the error sensors.

It is interesting to consider the scalar case of the LMS algorithm with such a modified cost function, which may be written in conventional vector form as

\[
\begin{align*}
\mathbf{w}_{n+1} &= \mathbf{w}_n - \alpha \left[ \mathbf{e}(n) \mathbf{X}_n + q \mathbf{y}(n) \mathbf{X}_n \right]. \\
\end{align*}
\]  

(36)

Substituting \( y(n) = \mathbf{X}_n^T \mathbf{w}_n \) gives

\[
\begin{align*}
\mathbf{w}_{n+1} &= \left[ \mathbf{I} - \alpha q \mathbf{X}_n^T \mathbf{X}_n \right] \mathbf{w}_n - \alpha \mathbf{e}(n) \mathbf{X}_n. \\
\end{align*}
\]  

(37)

In the case where \( \omega_y \) is \( \pi/2 \) and two coefficient filters are being used, the \( 2 \times 2 \) matrix \( \mathbf{X}_n^T \mathbf{X}_n \) will, on average, be equal to \( 1/2 \) multiplied by the identity matrix. The average behavior of the above algorithm is thus described by

\[
\begin{align*}
\mathbf{w}_{n+1} &= \gamma \mathbf{w}_n - \alpha \mathbf{e}(n) \mathbf{X}_n. \\
\end{align*}
\]  

(38)

where \( \gamma = 1 - \alpha q/2 < 1 \).

This implies that in the absence of any update term, the value of the filter coefficients would gradually decay away. This expression is exactly the same as that described by Widrow and Stearns [9, p. 377] as the "leaky LMS" algorithm.

B. Use of a Weighted Least Squares Criterion

It is sometimes desirable not to minimize the sum of the mean square values of a number of error signals, but to minimize the value of the largest one, the "minimax" criterion. In general, this minimization problem is very nonlinear and thus difficult to solve analytically. However, it has been suggested by Burrows and Shahinkaya [18] that a modified form of a least mean square solution could be used as an approximation to this, in which the weightings on the individual errors are varied depending on their mean square value. Burrows and Shahinkaya used an iterative matrix inversion formulation to solve their equations, and adjust their error weighting values after each iteration. They found that the algorithm converged after two or three iterations.

A similar approach can be taken in the stochastic gradient algorithm described by (35) if the effort weighting functions \( q_m \) are set equal to zero, and the error weighting functions are made equal to the averaged squared value of the relevant error signal

\[
\begin{align*}
p_i &= E\{\mathbf{e}_i^2\}. \\
\end{align*}
\]  

(39)

A simulation has been performed in which the mean square errors were approximated using two point moving averages and used to modify each \( p_i \), every sample, according to the equation

\[
\begin{align*}
p_i &= \left[ \mathbf{e}_i^2(n) + \mathbf{e}_i^2(n-1) \right]/2. \\
\end{align*}
\]  

(40)

The final results of this simulation were achieved after about 120 cycles, and it was found that the value of the largest error signal after adaption was 0.47, compared to a value of 0.68 with all values of \( q_i \) equal to unity. The sum of the mean square values of all the errors, however, increased to 1.08, from a value of 0.908 with all values
of $q_i$ equal to unity. In this simulation, the maximum mean square error has thus been reduced by about 30 percent, compared to the uniformly weighted sum of squares solution, at the expense of an increase in the total mean square error of about 17 percent. It should, however, be noted that after convergence the other error signals have a mean square value well below the maximum value of 0.47 quoted above. Consequently this is hardly a true minimax solution, which would drive the mean square values of all the errors to the same (minimum) value.

If the expression for $p_i$ in (39) is substituted into the original error criterion of (33) with all $q_{in} = 0$, it can be seen that the error criterion is the sum of the $L_2$ norms of the individual error signals. This is in contrast to the $L_2$ norms used in the normal stochastic gradient algorithm above and the $L_{\infty}$ norm which should be used in a true minimax criterion. In fact, higher order norms can be minimized by taking $p_i = E \{ (e_i)^p \}$, for example, which would eventually minimize the sum of the $L_{2k+2}$ norms of the individual error signals, as discussed in the single channel case by Walach and Widrow [19]. A practical problem associated with such algorithms is the very slow convergence rate due to the large difference in magnitudes of the individual terms of the coefficient update equation.

VI. CONCLUSIONS

A generalization of the filtered $x$ LMS algorithm has been presented which minimizes the sum of the mean square outputs of a number of errors, each linearly related to the outputs of a number of adaptive filters. The derivation of the algorithm involved the assumption that the adaptive filters were only varying slowly compared to the timescale of the response of the system to be controlled. However, simulations of the algorithm using a sinusoidal reference, and a practical implementation in an active sound control application, have shown that the algorithm is able to converge in a time comparable to the response time of the system to be controlled. The simulations of the algorithm also indicate that the total error converged to a value close to the optimum least mean square solution, and that it was robust to errors made in the assumed response of the system to be controlled and to uncorrelated measurement noise [11].

Similar behavior is also shown by a simplified, single input, single output version of the algorithm, which corresponds to the filtered $x$ LMS algorithm [9]. The pole positions of the equivalent transfer function, derived using the approach of Glover [14], can, however, be easily evaluated in this case. These can be used to analytically derive an expression for the optimum convergence coefficient, which, in this case, agrees well with computer simulations, and is approximately equal to the reciprocal of the delay in the system to be controlled, measured in samples. The equivalent transfer function can also be used to analytically demonstrate that there is a $\pm 90^\circ$ phase condition on the estimate of the system response in the limit of slow adaptation.

Two modifications to the multichannel algorithm are presented. The first which penalizes effort as well as error, and the second which minimizes the sum of a higher order function of the errors. The latter algorithm tends to a minimax solution as the power to which the individual errors are raised is increased.

APPENDIX

THE EQUIVALENT TRANSFER FUNCTION OF THE MULTIPLE ERROR LMS ALGORITHM WITH SINUSOIDAL REFERENCE

In this Appendix we use the approach developed by Glover [14] to obtain a matrix of equivalent transfer functions between the desired signals, considered as the inputs to the system, and the error signals, considered as the outputs of the system. It is found that if the reference signal is a synchronously sampled sinusoid, then the multichannel adaptive canceller behaves exactly like a linear, time invariant system between the desired and error signals. These transfer functions can thus be used to calculate the response of the system to any desired input excitation, and can also be used to investigate the stability of the algorithm by examining the positions of the poles of the transfer function.

The $m$th adaptive filter is fed from a reference signal to the form

$$x(n) = \cos (\omega_p n),$$

(A.1)

the output of which passes through a secondary path filter $C_{lm}$ before being summed and added to a desired signal to form the $l$th error signal, each of which is fed back to the adaptive filter. The algorithm used to adapt this filter is given by

$$w_{ml}(n + 1) = w_{ml}(n) - \alpha \sum_{i=1}^{L} e_l(n) \hat{r}_{lm}(n - i).$$

(A.2)

The filtered reference signal $\hat{r}_{lm}(n)$ is formed here by passing $x(n)$ through an estimate $(\hat{C}_{lm})$ of the true secondary path. By making this filter different from $C_{lm}$, the effect of errors in the estimate of the error path transfer function can be investigated.

The filter $\hat{C}_{lm}$ is, however, only excited by $x(n)$ at the reference frequency $\omega_p$. So if the modulus and phase of its transfer function at this frequency are

$$\hat{C}_{lm} (e^{j\omega_p}) = A_{lm} e^{j\Phi_{lm}},$$

(A.3)

the filtered reference signals must be

$$\hat{r}_{lm}(n) = A_{lm} \cos (\omega_p n + \Phi_{lm})$$

(A.4)

\[ \therefore \hat{r}_{lm}(n - i) = A_{lm}/2 \left[ \exp \left( j\Phi_{lm} + \omega_p i \right) \right. \]

\[ \quad \times \left. \exp \left( -j\omega_p n \right) \right] + \exp \left( -j\Phi_{lm} - \omega_p i \right). \]

(A.5)

In each of the update terms for the coefficient $w_{ml}$, this
signal is multiplied by $e_l(n)$. If the Z transform of $e_l(n)$ is $E_l(z)$, then the Z transform of the product $e_m(n) r_{lm}(n - i)$ is

$$Z\{e_l(n) r_{lm}(n - i)\} = \frac{A_{lm} L}{2} \left[ \exp \left\{ j(\Phi_{lm} - \omega_o i) \right\} E_l(z \exp(-j\omega_o)) \right.$$

$$\left. + \exp \left\{ -j(\Phi_{lm} - \omega_o i) \right\} E_l(z \exp(j\omega_o)) \right].$$

(A.6)

Taking the Z transform of the update equation, with $W_{mi}(z)$ as the Z transform of $w_{mi}(n)$,

$$W_{mi}(z) = -\frac{\alpha}{2} U(z) \sum_{i=1}^{L} A_{lm} \left[ \exp \left\{ j(\Phi_{lm} - \omega_o i) \right\} \right.$$  

$$\cdot E_l(z \exp(-j\omega_o))$$

$$\left. + \exp \left\{ -j(\Phi_{lm} - \omega_o i) \right\} E_l(z \exp(j\omega_o)) \right].$$

(A.7)

where $U(z) = 1/(z - 1)$. The output of the filter $y_m(n)$ is formed from

$$y_m(n) = \sum_{i=0}^{L-1} w_{mi}(n) x(n - i)$$

(A.8)

where

$$x(n - i) = \frac{1}{2}(e^{j\omega_0 n} e^{-j\omega_o i} + e^{-j\omega_0 n} e^{j\omega_o i}).$$

(A.9)

If we take the Z transform of each term in the summation for $y(n)$, we have

$$Y_m(z) = \frac{1}{2} \sum_{i=0}^{L-1} \left[ W_{mi}(ze^{-j\omega_o}) e^{-j\omega_o i} + W_{mi}(ze^{j\omega_o}) e^{j\omega_o i} \right].$$

(A.10)

Therefore,

$$Y_m(z) = -\frac{\alpha}{2} \sum_{i=1}^{L} A_{lm} \left\{ U(ze^{-j\omega_o}) \right.$$  

$$\cdot e^{-j\omega_o i} \left[ e^{j(\Phi_{lm} - \omega_o i)} E_l(ze^{-j2\omega_o}) \right.$$  

$$+ e^{-j(\Phi_{lm} - \omega_o i)} E_l(z) \right] + U(ze^{j\omega_o}) e^{j\omega_o i}$$

$$\cdot \left[ e^{j(\Phi_{lm} - \omega_o i)} E_l(z) \right.$$  

$$+ e^{-j(\Phi_{lm} - \omega_o i)} E_l(ze^{j2\omega_o}) \right\}. \right.$$

(A.11)

Therefore,

$$Y_m(z) = -\frac{\alpha}{2} \sum_{i=0}^{L-1} A_{lm} \left\{ \left[ E_l(z) \right] U(ze^{-j\omega_o}) \right.$$  

$$\cdot e^{-j\Phi_{lm}} + U(ze^{j\omega_o}) e^{j\Phi_{lm}} \right]$$

$$+ \left[ E_l(ze^{-j2\omega_o}) U(ze^{-j\omega_o}) e^{j(\Phi_{lm} - 2\omega_o i)} \right.$$  

$$+ E_l(ze^{j2\omega_o}) U(ze^{j\omega_o}) e^{-j(\Phi_{lm} - 2\omega_o i)} \right\}.$$

(A.12)

The equation in the second square brackets contains terms of the form $e^{\pm j2\omega_0 i}$ multiplied by a number of other terms which do not depend on $i$ and can thus be taken outside the summation. Evaluating the summation of these exponential terms, we obtain

$$\sum_{i=0}^{L-1} e^{\pm j2\omega_o i} = \frac{1 - e^{\pm j2\omega_o}}{1 - e^{j2\omega_o}}$$

$$= e^{\pm j2\omega_0 (1 - i)} \sin (\omega_o i) \sin (\omega_o) \right.$$

(A.13)

This is similar to the variable "$\beta$" discussed by Glover [14].

If the reference signal is synchronously sampled and the number of filter coefficients is equal to an integer $(k)$ multiplied by half the number of samples per cycle, then

$$I = k\pi/\omega_o \therefore \omega_o = k\pi/I$$

(A.14)

consequently, the second term in the square brackets in the summation above is identically zero. We are left with $I$ identical terms in $E_l(z)$, and substituting for $U(z)$ we obtain

$$Y_m(z) = -\frac{\alpha}{4} \sum_{i=1}^{L} A_{lm} E_l(z) \left[ \frac{e^{-j\Phi_{lm}}}{ze^{-j\omega_o} - 1} + \frac{e^{j\Phi_{lm}}}{ze^{j\omega_o} - 1} \right].$$

(A.15)

Therefore,

$$Y_m(z) = -\frac{\alpha}{4} \sum_{i=1}^{L} A_{lm} \left[ \frac{z \cos (\omega_o - \Phi_{lm}) - \cos \Phi_{lm}}{1 - 2z \cos (\omega_o) + z^2} \right] E_l(z)$$

$$= \sum_{i=1}^{L} G_{mi}(z) E_l(z), \text{ say.}$$

(A.16)

This may be written in matrix form

$$Y(z) = G(z) E(z)$$

(A.17)

where

$$Y(z) = \begin{bmatrix} Y_1(z) \ Y_2(z) \cdots \ Y_M(z) \end{bmatrix}^T$$

$$E(z) = \begin{bmatrix} E_1(z) \ E_2(z) \cdots \ E_l(z) \end{bmatrix}^T$$

and

$$G(z) = \begin{bmatrix} G_{11}(z) & G_{12}(z) & \cdots & G_{1M}(z) \\ G_{21}(z) & G_{22}(z) & \cdots & G_{2M}(z) \\ \vdots & \vdots & \ddots & \vdots \\ G_{M1}(z) & G_{M2}(z) & \cdots & G_{MM}(z) \end{bmatrix}.$$  

However, by generalizing the frequency domain formulation in Section II, we also have

$$E(z) = D(z) + C(z) Y(z)$$

(A.18)
Therefore, \[ E(z) = D(z) + C(z) G(z) E(z). \] (A.19)

For a given set of \( C_{lm}(z) \) and \( \hat{C}_{lm}(z) \), the elements of \( C(z) \) and \( G(z) \) could be evaluated and thus the stability of the multichannel algorithm could be determined, in principle, by examining the magnitude of the eigenvalues of the matrix \( [I - CG] \). However, even for the relatively simple system with two sources and four sensors used in \[ \text{[11]}, \] the characteristic equation is an 80th-order polynomial in \( z \). The determination of the coefficients of this polynomial would require a considerable amount of algebraic manipulation even before its roots are evaluated. Insight into the behavior of these algorithms can be gained, however, by considering the simplified case of a single channel system, as discussed in Section IV-B above.

\section*{Acknowledgment}

We would like to thank the numerous people who have influenced the preparation of this paper, in particular, P. Davies, J. K. Hammond, N. Gant, and A. R. D. Curtis.

\section*{References}

\begin{enumerate}
\end{enumerate}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image1}
\caption{Image description}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{image2}
\caption{Image description}
\end{figure}


After a period as a Research Fellow at the Institute of Sound and Vibration Research, University of Southampton, England, and as a temporary Lecturer in the Physics Department of the University of Surrey, he was appointed as a Lecturer in ISVR under the SERC Special Replacement Scheme. His research interests include the physical and signal processing aspects of speech and active control.

Ian M. Stothers received the Honours degree in engineering acoustics and vibration from the Institute of Sound and Vibration Research, University of Southampton, England, in 1985. Since then he has continued to work at the ISVR developing adaptive signal processing software and hardware, and applying this to acoustic and electroacoustic problems.

Philip A. Nelson received the B.Sc. degree in mechanical engineering in 1974 from the University of Southampton, England. He went on to undertake research into aerodynamic sound production at the Institute of Sound and Vibration Research and received the Ph.D. degree in 1981. From 1978 to 1982 he worked as a Research Development, and Consulting Engineer at Sound Attenuators, Ltd., Colchester, England, where he first became interested in the active control of sound. In 1982 he was appointed Lecturer at the Institute of Sound and Vibration Research, and has continued work on active techniques in acoustics and vibrations.